Describing Digital Signal Processing Systems in A Functional Reactive Style

Gábor Páli, Zoltán Gilián, Zoltán Horváth, and Anders Persson

Digital signal processing (DSP) often appears as a main component of specialized embedded systems, like radio base stations in the field of telecommunications. In such systems several discrete steps of DSP transforms are composed into applications to perform a complete function, simultaneously handling multiple flows of data with similar chains of processing configured dynamically and individually. Specifications for DSP systems are often written in a style that emphasizes a compositional style, and algorithms are described by themselves and their different configuration-dependent compositions in different documents. It is the responsibility of the application developer to assemble the different parts into a complete application. The majority of this problem consists of designing the interaction of the parts in the processing chain. Properties of the interaction include method of data transfer, data format, and spatial locality.

DSP algorithms or kernels within the system can be already efficiently described in Feldspar [4], a high-level domain-specific language, but currently there is no solution to describe their interaction and orchestration. As an attempt to fill this gap we propose an extension to Feldspar to address these problems. In practice DSP systems are essentially real-time domain-specific embedded systems with strict constraints on both resource usage and execution time. Our recent work, which is being carried out as part of a joint research project of Ericsson Research, Chalmers University of Technology (Gothenburg, Sweden), and Eötvös Loránd University (Budapest, Hungary), focuses on providing language and compiler support for scheduling multiply running instances of connected DSP kernels in different configurations with deadlines and providing resources for them in an embedded DSP environment. We present a potential design of such systems in a domain-specific style which is supported by efficient and reliable compilation techniques as we have learned in case of Feldspar [5] [6] and as others have shown for generic operating systems [3]. Our goal is to propose a design and a prototype implementation for the required language extensions and their compilation to C.

We have been modeling the current design flow at Ericsson in C, and have also been experimenting with using a notation similar to Haskell Arrows [1] in combination with elements of Functional Reactive Programming [2]. This paper presents the results of our experiments supported by case studies and examples in describing parts of radio base stations. The results can be used in future studies on how to design and implement a system-level layer above the current data-flow layer represented by Feldspar.

Acknowledgments

This work is being supported by Ericsson Software Research, Ericsson Business Network Units and SSF (Sweden), the Hungarian National Development Agency (KMOP-2008-1.1.2), Programul Operaţional Sectional Dezvoltarea Resurselor Umane 2007–2013 (POSDRU/6/1.5/S/3-2008, Romania), and the Swedish Research Council (grant no. 621-2009-5638).

References

[1] J. Hughes: Generalising Monads to Arrows, *Science of Computer Programming 37*, pp. 67–111, May 2000.

[2] H. Nilsson, A. Courtney, J. Peterson: Functional Reactive Programming, Continued, *Haskell Workshop 2002*, Pittsburgh, pp. 51–64, 2002.

[3] P-E. Dagand, A. Baumann, T. Roscoe: Filet-o-Fish: practical and dependable domain-specific languages for OS development. *In 5th Workshop on Programming Languages and Operating Systems (PLOS)*, Big Sky, MT, USA, October 2009.

[4] G. Dévai, M. Tejfel, Z. Gera, G. Páli, Z. Horváth: Feldspar – A Functional Embedded Language for Digital Signal Processing, *8th International Conference on Applied Informatics*, Eger, Hungary, January 27 – 30, 2010.

[5] G. Dévai, M. Tejfel, Z. Gera, G. Páli, Gy. Nagy, Z. Horváth, E. Axelsson, M. Sheeran, A. Vajda, B. Lyckegård, A. Persson: Efficient Code Generation from the High-level Domain-specific Language Feldspar for DSPs, *Workshop on Optimizations for DSP and Embedded Systems, in conjunction with IEEE/ACM International Symposium on Code Generation and Optimization (CGO 2010)*, Toronto, Canada, April 24–28, 2010,

[6] E. Axelsson, K. Claessen, G. Dévai, Z. Horváth, K. Keijzer, B. Lyckegård, A. Persson, M. Sheeran, J. Svenningsson, A. Vajda: Feldspar: A Domain Specific Language for Digital Signal Processing algorithms, *Eighth ACM/IEEE International Conference on Formal Methods and Models for Codesign*, Grenoble, France, July 26–28, 2010.